

REMARKS/ARGUMENTS

Claims 1, 3-10 and 32-50 remain in the application for further prosecution. By this amendment, claims 1, 32, 40, 47 and 49 have been amended. Claims 2 and 11-32 were canceled in the previous amendment.

Claim Rejections – 35 USC § 103

Claims 1 and 3-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Publication No. 2003/0188231 A1 (“Cronce”).

Claims 32-50 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Cronce, in view of U.S. Patent No. 6,487,301 (“Zhao”).

The present claims relate to an authentication method and system that allows for faster authentication of wagering game software by determining whether a particular predetermined amount of data is executable code or graphic data. The method and system only authenticates executable code, and not the graphic data during most runs thereby decreasing the authentication time over known systems that authenticate all parts of the wagering game software including generally larger graphic data.

A. Cronce Does Not Contemplate Selective Authentication For Game Code

Cronce relates generally to a security system that protects software from attacks by scattering the parts of a validation algorithm throughout the protected software program. (See Fig. 4a). Cronce addresses the problem of a standard validation routine that is locally situated and therefore can be circumvented by a hacker studying the code. (Abstract). Unlike the present claims, Cronce protects the entirety of the software for each authentication but does not determine whether a code block is either executable code or graphic data in order to speed authentication of more critical blocks, such as executable code, in comparison to less essential parts, such as graphic data. Cronce is more secure in certain aspects because it thwarts hackers

that could examine the code to find the validation routine and thus study this routine to tamper with the software to be protected. However Cronic does not speed the authentication of the gaming program in comparison to the methods and systems in the present claims. Cronic does review a software block to determine whether it contains the scattered checksum algorithm, but does not determine whether the software block is executable code or graphic data. Cronic discloses authenticating the entirety of the software and therefore teaches away from authenticating only parts of the software based on the type of code in each software block. Cronic is merely representative of the prior art that the present invention seeks to improve.

The Office Action has asserted that Cronic does not explicitly state that the application code contains both executable and graphic code but it would have been obvious to include both graphic code and executable code in the blocks of code. (p. 3). Even if this assertion were correct, as will be explained below, Cronic would teach away from the concepts of the claims because both the graphic code and executable code would be authenticated because all blocks are verified. Cronic specifically teaches that all of the code (other than the CRC validation routine that is embedded in the code) is authenticated.

B. Claims 1, 32 and 40 Are Allowable Over Cronic Because They Require Determination of Whether The Next Predetermined Amount of Data is Executable Code Or Graphic Data

Initially, claims 1, 32 and 40 are allowable over Cronic (either alone or in combination with Zhao) because they require “determining if the next predetermined amount of data is executable code or graphic data” such as in claim 1. As explained above, Cronic only determines whether a software block relates to the validation routine, but does not determine any other characteristics of the software block to be validated. The Office Action cites Fig. 3b, element 310, for teaching the claimed feature of determining if the next amount of data is

executable code or graphic data. (p. 3). Applicant respectfully disagrees that this or any other section of Cronicc discloses determining whether the next amount of data is executable code or graphic data. Fig. 3b in Cronicc generally relates to a process used by a tool to prepare a modified application for a run-time checksum validation algorithm. The tool scatters the checksum algorithm in the software itself. (paragraphs 28 and 39). Element 310 is described as where the prepared software application is received for processing. (paragraph 30). The routine simply checks for exported symbols for the start of the blocks to be protected by the checksum algorithm. (paragraph 30). There is nothing in this paragraph or any other description of Fig. 3b to indicate that the software is checked to determine whether it is executable code or graphic data. Indeed, as long as the software block has a corresponding checksum, it is authenticated whether it is executable code or graphic data. (paragraph 51). There is no disclosure in Cronicc for determining whether the code is executable code or graphic data, nor is there any suggestion why one of skill would consider the separation between the types of code used in the gaming program.

C. The Claims Are Separately Allowable Because They Require Selective Authentication Of Executable Code

Applicant has amended independent claims 1, 32, 40 and 47 to require that the executable code operates a wagering game on the gaming machine and the graphic data is accessed by the executable code to display wagering game graphics on a display. The Office Action has asserted that it would be obvious to include graphic data and executable code together in blocks of code. (p. 3). The Office Action cites paragraph 30 of Cronicc as disclosing that executable code will be subject to verification and graphic data will not be verified. (p. 3). Applicant respectfully disagrees with this assertion. At best, paragraph 30 discloses “defining the start and end of program blocks to be protected by runtime checksum validation.” If the Office Action is correct

and it is obvious that such program blocks may contain both executable code and graphic data, both executable code and graphic data would be validated by the Cronic system.

In contrast, claim 1 requires that if the predetermined amount of code is graphic data, the next predetermined amount of data is read and if the next predetermined amount of data is executable code, then said executable code is authenticated. Similar elements are present in claims 32, 40 and 47. Cronic also does not disclose these elements because Cronic requires authenticating the entire code ("each block"), regardless of the code type. (paragraph 51). As explained above, nothing in this paragraph or anywhere else in Cronic relates to authenticating the software block if the block is executable code and not authenticating by reading the next block if it is not executable code. In fact, as long as the block is not part of the checksum algorithm, it is validated and therefore Cronic actually teaches away from the Applicant's claims, which require that graphic data is not always authenticated when the executable code is authenticated.

Cronic may be further distinguished because it discloses that the only parts of the program block that are not authenticated are the checksums and authentication routine which are not involved with the operation of the software (wagering game) in any way. The amended claims now clearly require that the executable code operates the wagering game and the graphic data is accessed by the executable code. Because Cronic's checksum routine is not executable code for operating a wagering game or graphic data to display wagering game graphics, Cronic does not disclose authenticating executable code for operating a wagering game but not authenticating graphic data accessed by the executable code to display wagering game graphics on a display.

D. Independent Claims 32, 40 and 47 Are Separately Allowable over Cronic and Zhao Because The Combination of Cronic and Zhao Do Not Authenticate Graphic Data Separately From Executable Code

With regard to claim 32-50, the Office Action has combined Cronic's software authentication process with the graphic data authentication process disclosed by Zhao. (pp. 6-7). The Office Action asserts that one of skill would be motivated to authenticate graphic data to prevent a person from copying a digital representation without degradation. (p. 7). Initially such a combination would not be made by one of ordinary skill in the art because Cronic (and the current application) relates to authentication of digital data (software) while Zhao relates to an entirely different authentication issue, namely authenticating a physical graphic such as a fax or a printed document. Zhao assumes that an image is scanned into an analog form and includes a watermark that is transferred in the scan. (Col. 7, ll. 25-47). Zhao is not directed toward authentication of the actual electronic data and therefore the combination of Cronic and Zhao would require the entirety of the wagering game graphics to be printed out to apply Zhao's technique of visual authentication. This combination would be clearly unworkable in the wagering game environment which requires rapid authentication.

However, even combining Zhao with Cronic would neither disclose nor suggest authenticating executable code and only authenticating graphic data when a predetermined condition has been met as required by claim 32. Even assuming Zhao discloses a method of authenticating graphic data in a software program, Zhao is only limited to authentication of the physical graphic itself. As explained above, Cronic does not disclose or suggest authenticating executable code and only authenticating graphic data under certain conditions. Cronic discloses authenticating all of the code at the same time with the same authentication routine, regardless whether the code is executable code or graphic data. Therefore the combination of Cronic and

Zhao would authenticate executable code and graphic data during the same authentication cycle.

Claim 32 and its dependents are therefore allowable over any combination of Cronic and Zhao.

Similarly, claim 40 is allowable over Cronic and Zhao. Cronic does not disclose authenticating the executable code and authenticating graphic data only if a predetermined condition is met as required by claim 40. The combination of Cronic and Zhao would not suggest or teach such an element because Cronic would simply authenticate the software regardless of whether it is executable code or graphic data. Claim 40 and its dependents are therefore allowable over the cited references.

Claim 47 requires “authenticating said executable code at a first frequency and authenticating said graphic data at a second frequency, said first frequency being greater than said second frequency.” Neither Cronic nor Zhao disclose or suggest authenticating parts of the code at different frequencies. Zhao is silent as to authentication of executable code or the frequency at which different types of code are authenticated. Cronic actually teaches away from authenticating different code types of a software program at different frequencies because Cronic discloses authenticating all parts of the program at the same time, i.e. at the same frequency. (paragraph 51). Claim 47 and its dependents are therefore allowable over the cited references.

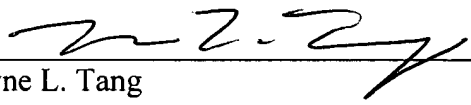
Conclusion

It is Applicant's belief that all of the pending claims 1-10 and 32-50 are now in condition for allowance and actions towards that effect is respectfully requested.

If there are any matters which may be resolved or clarified through a telephone interview, the Examiner is respectfully requested to contact the undersigned attorney at the number indicated.

Respectfully submitted,

Date: July 23, 2008



Wayne L. Tang
Reg. No. 36,028
NIXON PEABODY LLP
161 N. Clark Street, 48th Floor
Chicago, Illinois 60601-3213
(312) 425-3900
Attorney for Applicants